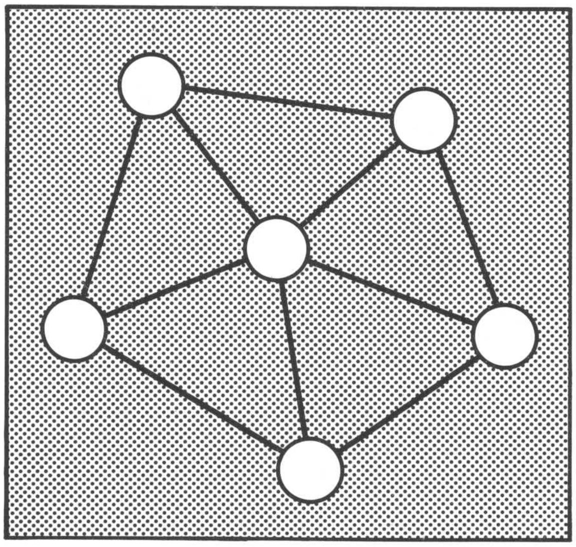
BRE421/D

A SHORT TUTORIAL ON NETWORKING



MOTOROLA

CONTENTS:

CHAPTER 1

THE O.S.I. MODEL AND THE SEVEN LAYER

1.1 INTRODUCTION

A Computer network is a very complex system allowing different computer hosts to communicate among themselves over long distances.

Most of the time, if host A wants to communicate with host B there will be no permanent physical cabling between them. Instead the situation will be very similar to a telephone link: computer A will be connected to a specialised communication computer called "Intermediate Message Processor" or **IMP**, which will handle the communication task. The IMP itself will be linked to other similar IMPs, the last one being connected to host B.

The set of all IMPs or nodes involved is called a subnet (Fig. 1.1).

The link between a,b,c,d... does not need to be permanent. As in a telephone network it can be a switched link established only for the time it will be needed. Making host A communicate with host B can be very difficult, especially if A and B come from different vendors. The only reasonable way to achieve this is to partition it into different subtasks using a model to describe the network.

The International Standards Organisation **(I.S.O.)** has proposed a general model to describe network architectures. This is called the Open Systems Interconnection model **(O.S.I.)**. Every article published today refers to the OSI model, so it is necessary to describe its' main aspects.

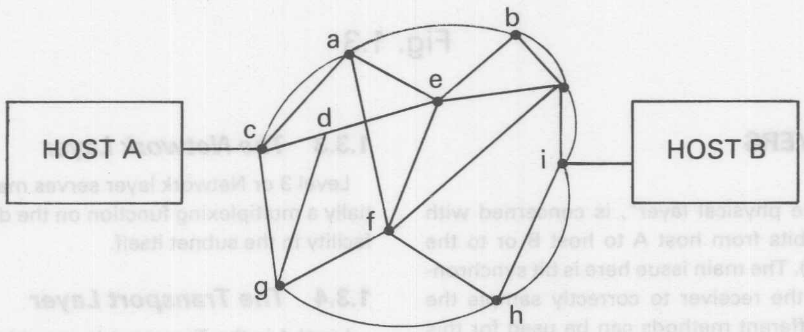


Fig. 1.1

1.2 THE OSI MODEL

The O.S.I. model classifies the tasks involved in a computer network in seven layers arranged in an hierarchical order. It is represented on fig. 1.2.
Each layer has a name which corresponds to its function.

For each layer attached to host A there is a corresponding layer attached to host B, these two layers are called **"peer layers"**. Although the material exchange of bits is done only on the physical layer, any layer should be considered as being able to communicate with its peer via a logical link.

The communications between peer entities must follow a set of rules and conventions which altogether constitute a **"protocol"**.

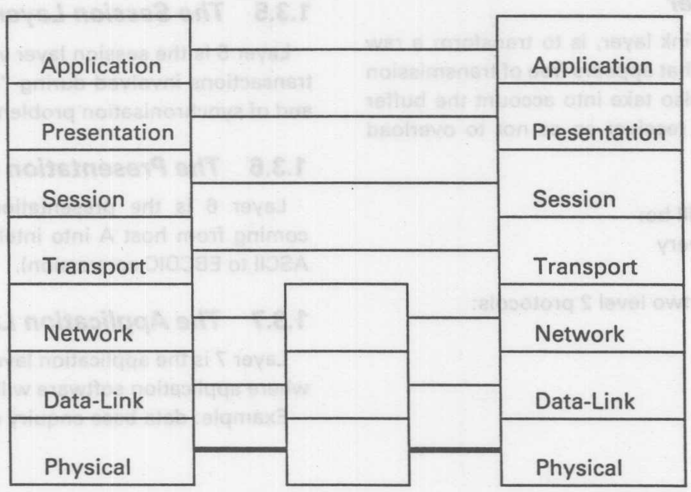


Fig. 1.2

To better grasp this concept of peer-to-peer communication one may think of two state leader, Ronald and Nicolai, wanting to exchange some information via their ambassadors. (Fig. 1.3).

President Ronald will pass his message to his ambassador Arthur who will deliver it to his "peer" ambassador Petrov, who will forward it to President Nicolai.

In this exchange there has been no physical meeting of the two presidents, but the message has nevertheless made its' way from Ronald to Nicolai.

One can naturally extend this description to include any number of intermediate layers.

Fig. 1,2 shows that for the first three layers there is an intermediate box between host A and host B. These layers must take into account the actual exchange of information between the host and the subnet. This exchange will be transparent for the above layers: 4,5,6 and 7.

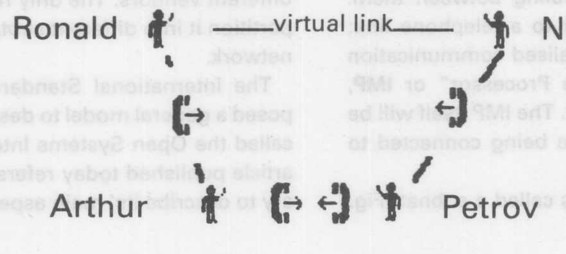


Fig. 1.3

1.3 THE SEVEN LAYERS

1.3.1 *The Physical Layer*

Layer 1, also called "the physical layer", is concerned with transmitting a stream of bits from host A to host B or to the intermediate node (or IMP). The main issue here is bit synchronisation, (or the ability of the receiver to correctly sample the incoming bits.) Several different methods can be used for this purpose:

Asynchronous transmission: Bit start, Bit stop
Synchronous transmission: Character oriented (Bisync)
Byte count oriented (DDCMP)
Bit oriented (HDLC, SDLC)

These techniques are described in detail in MOTOROLA's "Digital Data Communication Guide" ref. BRE244 hereafter referred as DDCG.

1.3.2 *The Data-Link Layer*

The task of layer 2, or data-link layer, is to transform a raw transmission facility into a link that appears free of transmission errors to layer 3. Layer 2 will also take into account the buffer capabilities of the sender and receiver so as not to overload them.

The main functions of level 2 will be:
- Error Checking and Recovery
- Flow Control

We will return in more detail to two level 2 protocols:
LAPB and SDLC

1.3.3 *The Network Layer*

Level 3 or Network layer serves many purposes, but is essentially a multiplexing function on the data-link level and a routing facility in the subnet itself.

1.3.4 *The Transport Layer*

Level 4 is the Transport layer which takes care of the logical communication between end users. The way in which the messages have been conveyed through the network is irrelevant for layer 4.

The task of layer 4 will be to adapt message sizes to the capacity of layer 3 service and to make sure that on arrival messages are reconstituted in their integrity before they are passed to the upper layer.

Now that the task of transfering good quality messages from host A to host B is fulfiled, let us turn our attention to the higher layers 5, 6, and 7.

1.3.5 *The Session Layer*

Layer 5 is the session layer which mainly takes care of all the transactions involved during "login" and "logout" procedures and of synchronisation problems.

1.3.6 *The Presentation Layer*

Layer 6 is the presentation layer which transforms data coming from host A into intelligible data for host B (example ASCII to EBCDIC conversion).

1.3.7 *The Application Layer*

Layer 7 is the application layer. This is now the user's domain where application software will be running.

Example: data base enquiry etc.

CHAPTER 2

THE PHYSICAL LAYER

2.1 DATA TRANSMISSION TECHNIQUES

Before going into detail for each of the seven layers, let us recall some fundamentals about digital data transmission.

There are essentially two different techniques used for digital data transmission:

Asynchronous data transmission
Synchronous data transmission

These two techniques are described in detail in MOTOROLA's DDCG.

Asynchronous transmission is so called because the receiver uses a local clock to recover the incoming bits. In this mode, only one ASCII character is sent at a time, embedded between start and stop bits which are used by the receiver for synchronisation. This mode is generally used for slow terminals only: as its' efficiency is poor and there is almost no error tracking except for parity.

Synchronous transmission: here the clock signal must be transmitted along with the data, or regenerated on the reception side from the incoming data. All synchronous protocols include a mechanism for error detection and recovery.

Let us summarize the main types of synchronous protocols.

– **Character sequence oriented**: uses specific sequences of data characters from an alphanumeric code.

Example: AT&T Selective Calling System 83 B

– **Control character oriented:** uses specific control character from an alphanumeric code

Example: IBM Binary Synchronous Protocol (Bisync)
ANSI Data Link Control X 3.28
ISO Basic Mode Control Procedure

– **Byte count oriented:** uses a header which includes a count of the number of data bytes to be sent and other information.

Example: DEC Digital Data Communication Message Protocol

– **Bit pattern oriented**: uses special flags to delimitate the message and bit patterns to define control functions.

Example: IBM Synchronous Data Link Control (SDLC)
ANSI Advanced Data Communication Control Procedure (ADCCP) X 3.66
ISO High Level Data Link Control Procedure (HDLC)
Burroughs Data Link Control (BDLC)
Univac Universal Data Link Control (UDLC)

Character oriented protocols are an improvement over asynchronous bit start, bit stop protocols but still present a number of drawbacks. Among which:–

- Hardware and software must distinguish between data and control characters from the same code.
- Link control characters reduce the number of available data characters in the code.
- Difficult procedure for data transparency.
- Inherently half duplex.
- Text character can easily be changed by noise into control character.

This is why there is a clear evolution towards a generalisation of bit pattern oriented protocols.

2.2 DATA ENCODING AT THE PHYSICAL LAYER

Before the binary data is sent on the transmission line, it is generally encoded in order to better match the line characteristics. Fig. 2.1 shows the fundamental encoding techniques and Fig. 2.1b represents the binary data before encoding. As it is a sequence of 0's and 1's, the average current flowing in the line will not be zero.

One can improve this by using a "non return to zero" **NRZ** coding that greatly reduces the average current. (c)

If the clock signal has to be recovered from the data, **NRZI** or Non Return to Zero Inverted coding can be used: during long zero sequences the signal changes its' state for each new bit cell. (d)

In order to completely eliminate the DC component of the encoded signal, "biphase level", generally known as **"Manchester"** is often used. This technique is very easy to implement as it is only the result of an exclusive-or between clock and data signals. Fig. 2,2 shows that the power spectrum of the resultant signal has no DC component. (e)

Manchester encoding has one drawback: if the two wires of the data signal are crossed, the digital information can be altered. To solve this problem one can use **"differential bi-phase"** encoding which keeps a memory of the last transition. (f)

Fig. 2.1g shows a **"Miller"** encoding scheme in which one transition out of every two is suppressed compared to Manchester encoding.

Fig. 2.2 shows the power spectrum of the signal obtained. It is very narrow but has some DC component.

Fig. 2.1h represents **"pseudo-ternary"** encoding, which is specified for **I.S.D.N.** Here a "one" is represented by a zero level signal, and a "zero" is represented by a positive going or negative going pulse (depending on the last transition).

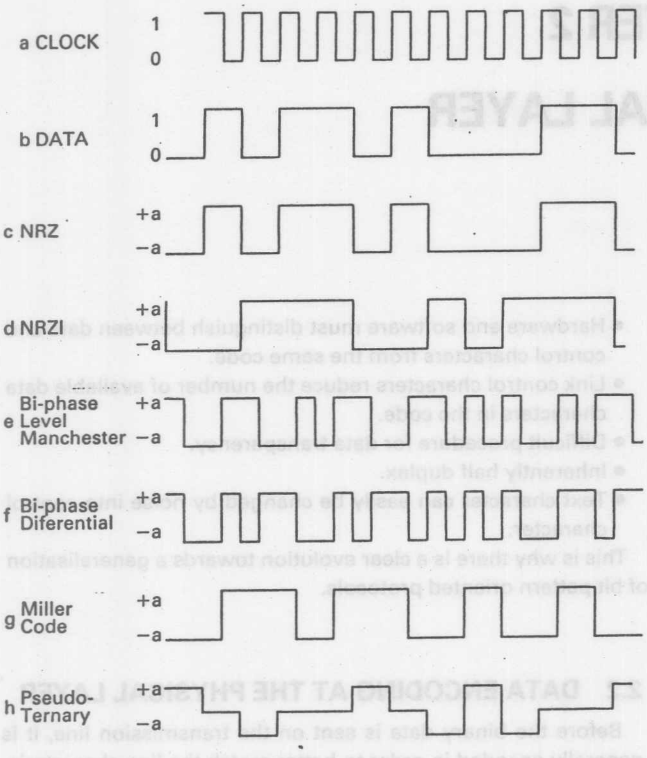


Fig. 2.1

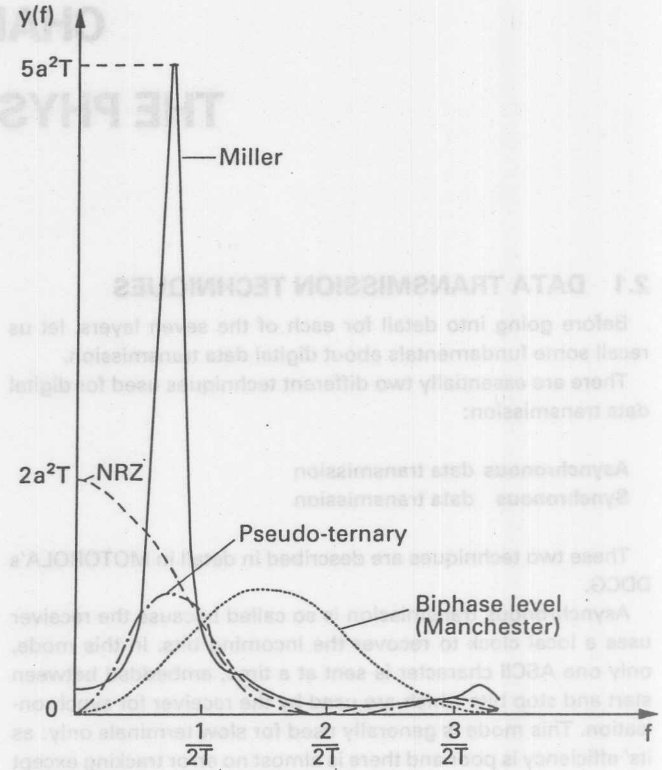


Fig. 2.2

2.3 RS 232.C INTERFACE

The Electronic Industries Association (**E.I.A.**) has developed a standard for the interface between the data communication equipment (**DCE**) and the data terminal equipment (**DTE**). The last version of this standard, RS232.C, issued in 1969 has gained universal acceptance.

The RS232 standard basically defines three specific items:–

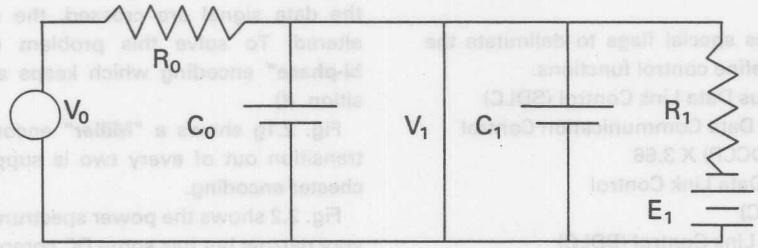
1) Electrical signal characteristics.
2) A functional description of the interchange circuits.
3) A list of standard subsets of specific interchange circuits for specific groups of communication system applications.

The RS232.C interface has the following area of application:–

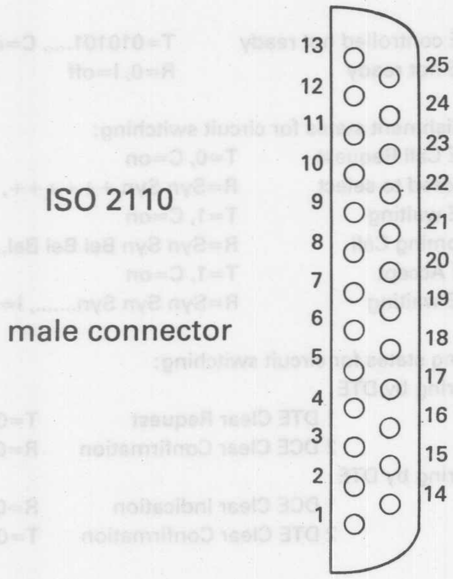
* Data signalling rates up to 20 kbytes
* Synchronous or Asynchronous transmission
* Leased line or dial-up network service
* Half-duplex or full-duplex transmission

INTERCHANGE EQUIVALENT CIRCUIT

Each lead can be modelled according to the following circuit



V_0: Open Circuit driver voltage
R_0: Driver internal resistance
C_0: Total effective capacitance associated with the driver
V_1: Voltage at the interface point
C_1: Total effective capacitance associated with the terminator
R_1: Terminator load DC resistance
E_1: Open circuit terminator voltage



2.4 CCITT RECOMMENDATIONS V.24/V.28 AND X.21 BIS

This is the CCITT version of RS232.C for data transmission over the public telephone network and the public data network.

V.24 defines the interchange circuits.
Series 100: general application
Series 200: automatic calling

V.28 specifies electrical characteristics of interchange circuits and is equivalent to RS232.C

2.5 EIA RS422/423/449 STANDARDS

–Replaces EIA RS232 for high speed data links.
–Accommodates advances in integrated circuit design, reduces crosstalk between circuit to allow greater distance between DTE and DCE, has higher data rates and new functions.
–**RS 422** specifies the electrical characteristics of balanced voltage digital interface circuits.
–**RS 423** specifies the electrical characteristics of unbalanced voltage digital interface circuit. (compatible with RS232)
–**RS 449** specifies the functional and mechanical characteristics of the DTE-DCE interface.
–These specifications correspond to the CCITT **V.10** and **V.11** (Telephone Network) and **X.26,X.27** (Data Network).

2.6 CCITT RECOMMENDATIONS X.20 AND X21:

* RS 422 and RS 423 standards offer a quality improvement over RS 232 and an increase of the pin count.
* CCITT Recommendations X.20 and X.21 represent an evolution of the data communication interface towards more logic and fewer pins, (only 15).
* X.20 and X.21 are more complex than the RS 232.C interface as they include some exchange of specific messages during link set-up or link disconnect phases.

X.20 INTERFACE:

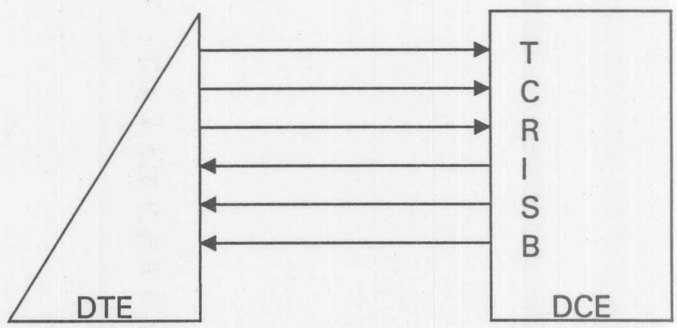
X.20 is CCITT defined as the "Interface Between Data Terminal Equipment (DTE) and Data Circuit Terminating Equipment (DCE) for Start-Stop Transmission Services on Public Data Networks."

The interface consists of the following signals:
G: Signal Ground
Ga: DTE Common Return (from DTE to DCE)
Gb: DCE Common Return (from DCE to DTE)
T: Transmit (from DTE to DCE)
R: Receive (from DCE to DTE)

X.21 INTERFACE:

X.21 is CCITT defined as the "Interface Between Data Terminal Equipment (DTE) and Data Circuit Terminating Equipment (DCE) for Synchronous Operation on Public Data Networks".

The Interface consists of the following signals:
G: Signal Ground
Ga: DTE Common Return (from DTE to DCE)
T: Transmit (from DTE to DCE)
R: Receive (from DCE to DTE)
C: Control (from DTE to DCE)
I: Indication (from DCE to DTE)
S: Signal Element Timing (from DCE to DTE)
B: Byte Timing (optional) (from DCE to DTE)



Main Characteristics:
–Six interchange circuits (2 wires) defined by Rec. V.24.
–International Alphabet no. 5 (ASCII), character control.
–Full bit sequence transparency.
–DTE must implement electrical characteristics of either X.27 (RS 422) or X.26 (RS 423).
–DCE must implement electrical characteristics of X.27.
–Standard 15 pin connector instead of 25 or 39 pin connectors defined by RS 232 or RS 449.
–Compatibility between X.21 and RS 449 achieved through simple adaptor.
–Reduction in number of functions: no separate send and receive timing circuits, no data signalling rate selection and no selection of standby facilities.
–No provision for secondary channel handling.

SOME USEFUL DEFINITIONS:

For circuits T and R:
- state 0: at least 15 consecutive 0's
- state 1: at least 15 consecutives 1's

For circuits C and I
- state ON: continuous 0's
- state OFF: continuous 1's

Four operational phases:
- ★ quiescent
- ★ call establishment
- ★ data transfer
- ★ clearing

Quiescent states for circuit switching:

1: DTE ready	T=1, C=off
2: DCE ready	R=1, I=off
3: DTE uncontrolled not ready	T=0, C=off
4: DTE controlled not ready	T=010101...., C=off
5: DCE not ready	R=0, I=off

Call Establishment states for circuit switching:

1: DTE Call Request	T=0, C=on
2: Proceed to select	R=Syn Syn ++++++, I=off
3: DTE waiting	T=1, C=on
4: Incoming Call	R=Syn Syn Bel Bel Bel...., I=off
5: Call Accept	T=1, C=on
6: DCE waiting	R=Syn Syn Syn......, I=off

Call Clearing states for circuit switching:

– Clearing by DTE

1 DTE Clear Request	T=0, C=off
2 DCE Clear Confirmation	R=0, I=off

– Clearing by DTE

1 DCE Clear Indication	R=0, I=off
2 DTE Clear Confirmation	T=0, C=off

CHAPTER 3

THE DATA LINK LAYER

3.1 INTRODUCTION

In this chapter, we will describe Station configuration, **HDLC** main features with a special emphasis on **LAPB**, and **SDLC**, the IBM data link protocol.

3.2 STATION CONFIGURATION

Before entering the fundamentals of the HDLC protocol, we must review the terminology used to describe various possible configurations.

HDLC is used for various applications: host-to-host communication, host-to-terminal, host-to-modem, point-to-point, multidrop etc... In any case the ends of the data-link are called "**stations**". Three different configurations must be differentiated:–

The unbalanced link
The symmetrical link
The balanced link

3.2.1 *Unbalanced Link*

In this case, a master station called "primary" communicates with one or several slave stations called "secondaries".

The primary station has absolute control over the link; no secondary station has the right to transmit before being authorized by the primary.

The primary station uses control frames called "**commands**", and the secondary station uses "**responses**".

In this configuration data-flow can only be half-duplex.

3.2.2 *Symmetrical Link*

A symmetrical link is a point to point link in which each station is composed of two sub-stations: a primary and a secondary. There is no permanent master and each station can in turn send commands or responses.

3.2.3 *Balanced Link*

In a balanced link each station is a combined primary and secondary. This is the configuration to which LAPB protocol applies. Here data-flow can be full-duplex.

3.3 HDLC: HIGH LEVEL DATA LINK CONTROL

HDLC is an international standard whose complete definition may be found in REF(1), but these are the main features.

3.3.1 *Frame Structure*

For HDLC protocol, bits are transmitted in sequences organised in frames. The general frame format is discussed in MOTOROLA's DDCG. We can see in Fig 3.4 that a frame comprises six different "fields". Each frame begins and ends with special bit pattern 01111110 ($7E) called a "flag".

The next field is a byte long address field. It will contain the destination address for a command or the source address for a response. Next comes the control field, one or two bytes wide depending on the number of outstanding frames specified which will be discussed at length later. Next comes the information field which can in principle be of any length although most of the time it will be an even number of bytes. Then comes the FCS (Frame Check Sequence) which will be checked by the receiver for data transmission integrity. Finally, there is a closing flag identical to the opening flag.

3.3.2 *Station State*

Before we look at the meaning of the different possible control fields (commands or responses), we must note that a given station at a given time can only be in one of the following three logical states:

– Disconnected
– Initialisation
– Transfer

The control field will then contain three types of commands:

1. – **Information commands** which inform the destination that the following field is an information field and carries N(R) and N(S) values. (The meaning of these parameters will be explained later).
2. – **Unnumbered commands** and **responses** used to set or disconnect the link and also to reject bad frames in case of errors which are recoverable by retransmission.

Unnumbered commands are:

SABM (Set Asynchronous Balanced Mode)
DISC (Disconnect the link)

Unnumbered responses are:

DM (Disconnect mode)
UA (Unnumbered Acknowledgement)
FRMR (Frame Reject)

3. – **Supervisory commands** and **responses** are used for the following purposes:

* Frame acknowledgement and indication of busy or not busy receiver state.

RR (Receiver Ready)
RNR (Receiver Not Ready)

* Rejection of out-of-sequence frames.

REJ

I FRAME FORMAT 2 INFORMATION	I FRAME FORMAT 2 INFORMATION
S FRAME FORMAT RR –RECEIVE READY RNR –RECEIVE NOT READY REJ –REJECT SREJ–SELECTIVE REJECT	S FRAME FORMAT RR –RECEIVE READY RNR –RECEIVE NOT READY REJ –REJECT SREJ–SELECTIVE REJECT
U FRAME FORMAT SNRM –SET NORMAL RESPONSE MODE SNRME (EXTENDED) SABM –SET ASYNCHRONOUS SABME–BALANCED MODE (EXTENDED) SIM SET INITIALISATION MODE TEST TEST UI UNNUMBERED INFORMATION XID –EXCHANGE IDENTIFICATION DISC –DISCONNECT	U FRAME FORMAT UA –UNNUMBERED ACKNOWLEDGEMENT DM –DISCONNECTED MODE TEST TEST UI UNNUMBERED INFORMATION XID –EXCHANGE IDENTIFICATION DISC –REQUEST DISCONNECT

3.3.3 *Flow and Error Control*

Full duplex transmission is achieved in HDLC protocol by using an acknowledgement mechanism that allows up to 7 (modulo 8) or up to 127 (modulo 128) information frames to be outstanding (not acknowledged). This number is called the **"window"** size. As new information frames are received and acknowledged on the receiver side, the window is advanced and the sender is allowed to send new frames.

Every station in the information transfer state maintains a send variable associated to the Information frame that is transmitted, and a receive variable associated to the Information frame that has been received.

The send variable is **V(S)** and the receive variable is **V(R)**. A control field preceding an information field will carry two parameters:

– **N(R)** acknowledges all the frames received up to N(R)−1, it is the sequence number of the next expected frame.
– **N(S)** is the sequence number of the frame which is currently being sent.

A **supervisory control field** carries only the parameter N(R), as an acknowledgement of the last correct frame received.

Upon receiving N(R) the transmitting station knows that all frames sent up to N(R)−1 have reached their destination. N(S) is the number of the frame being sent (i.e. the frame in the information field).

In this manner, the receiving station can check if the received frame sequence number N(S) is the one expected, (N(S)=V(R)). If not, the received frame is out of sequence and will be rejected.

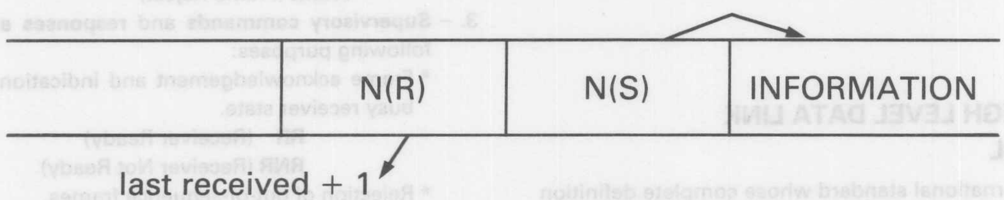
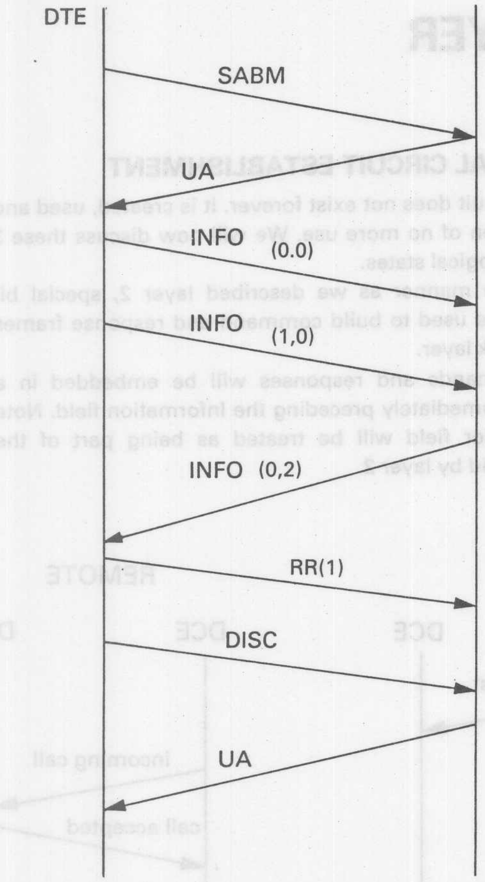


Fig 3.6 shows an example of a session between two hosts using HDLC LAPB protocol.



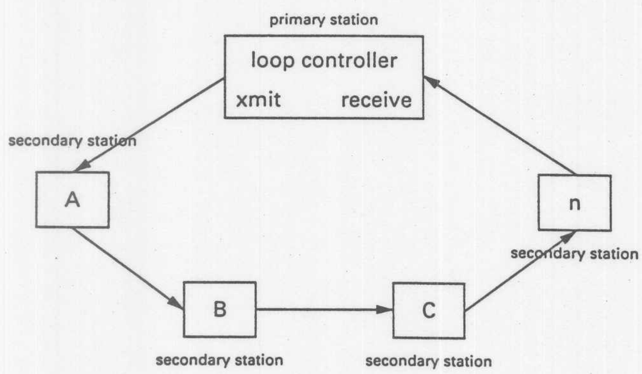
3.4 SDLC: SYNCHRONOUS DATA LINK CONTROL:

We will discuss only the aspects of SDLC which are different from HDLC. SDLC has some specific features not found in HDLC, namely:

- a loop mode
- non-sequenced information frames

3.4.1 *SDLC Loop-Mode*

In a loop, a one way communication channel originates at the transmitting part of the primary station, connects one or more secondary stations in a serial fashion and then terminates back at the receiver part of the primary station.



LOOP OPERATION:

The loop operation is logically operated as a half-duplex data-link in a loop configuration with only one station transmitting at a time: the primary station or one of the secondary stations. The secondary stations transmit sequentially as required by their order on the link.

3.4.2 *Primary Station Transmitting*

The primary station sends command frames that are addressed to any of the secondary stations on the loop. Each frame transmitted by the primary carries the address of the secondary station or stations to which the frame is directed.

Every secondary station on the loop decodes the address field of each frame transmitted by the primary station and serves as a repeater for all primary transmissions to down-loop stations.

When the primary has finished transmitting frames, it follows the last flag with a minimum of eight consecutive 0's (a flag followed by eight 0's is a "turn around" sequence. It then transmits continuous 1's which create a "go-ahead" sequence (01111111).

In this way, the primary controls all loop communications. The primary, while continuing to transmit 1's, goes into the receive mode.

3.4.3 *Secondary Station Transmitting*

Before transmitting in the loop, a secondary station must have received a frame addressed to it with the P bit set, or received a UP command with the P bit reset.

The first down-loop secondary detects the go-ahead sequence. If the secondary has a response to send, it changes the seventh 1 bit to a 0 bit, creating a flag pattern. It follows the flag with a response frame that contains an individual address. Following its' last frame it then again becomes a repeater, forwarding the continuous bits it receives from the primary station.

The next down-loop secondary operates similarly when it detects the go-ahead sequence which results from the continuous 1 bit pattern.

This procedure continues until the last down-loop secondary to transmit completes its' transmission.

CHAPTER 4

THE NETWORK LAYER

If we refer to the I.S.O. model, layer 3 is generally referred to as the Network Layer. This definition can be sometimes misleading as the concept of a layer 3 service can be defined without explicit reference to a network facility.

What will be the role of layer 3, i.e. what service does it provide to the next upper layer the transport layer?

4.1 NETWORK LAYER SERVICES

We already know that the service provided by layer 2 to layer 3 is a reliable data-path with flow and error control. The next step in complexity will be to share this data-path among several users or processes.

Thus the main service offered by layer 3 will be multiplexing. Data belonging to different processes will be packaged into fixed size packets (this is part of level 4's job), each packet will be labelled and layer 3 will ensure that any particular packet will be delivered with no data loss at the other end of the link.

4.2 VIRTUAL CIRCUITS

The modelisation of the multiplexing of packets on the data link uses the concept of virtual circuit, a logical association that exists between two host processes for a period of time. CCITT X.25 recommendation defines two types of virtual circuits: switched and permanent. A permanent virtual circuit is established by agreement with the public packet network administration, once created it is capable of sending and receiving data at all times.

A switched virtual circuit is established by a call set-up procedure and terminated by a call clearing procedure specified in the protocol.

A virtual circuit can be viewed as being composed of three elements:–

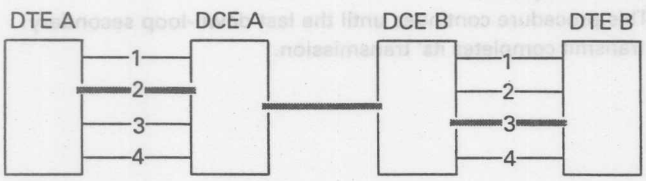


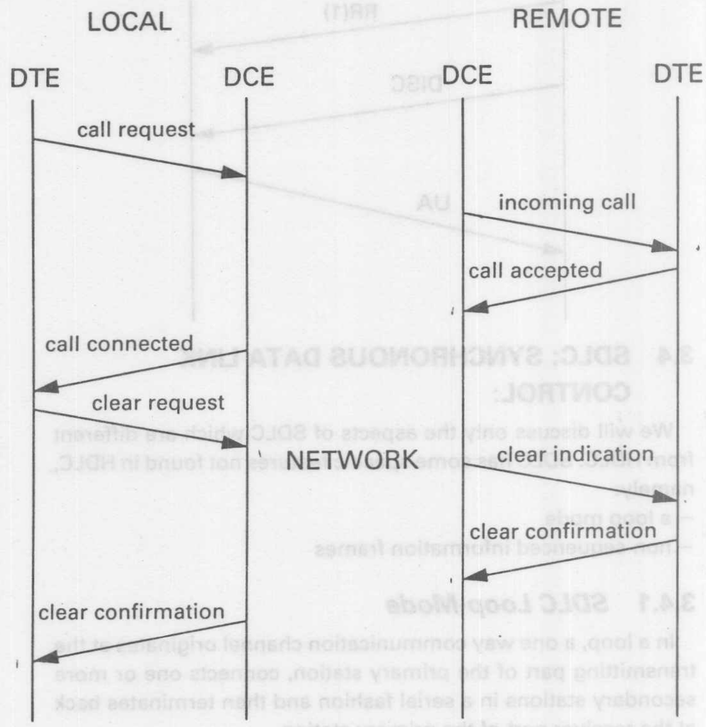
Fig 4.1 shows a virtual circuit established between DTE A and DTE B, made of a link (labelled 2) between DTE A and DCE A, a link between the local DCE and the remote DCE B (provided by the Network and outside the scope of Standards) and finally a link between the remote DCE B and the remote DTE B (labelled 3).

4.3 VIRTUAL CIRCUIT ESTABLISHMENT

A virtual circuit does not exist forever. It is created, used and destroyed when of no more use. We will now discuss these 3 virtual circuit logical states.

In the same manner as we described layer 2, special bit patterns will be used to build command and response frames for the Network layer.

These commands and responses will be embedded in a header field immediately preceding the Information field. Note that this header field will be treated as being part of the information field by layer 2.



CHAPTER 5

THE TRANSPORT LAYER

Let us now turn our attention to the TRANSPORT LAYER.

The lowest three layers were concerned with the transmission, framing, multiplexing and possibly routing of data-packets between the local DTE and DCE.

The transport layer will be in charge of providing a reliable end-to-end path to process data transport. We shall now examine what must be done to achieve this goal.

5.1 TRANSPORT LAYER SERVICES:

5.1.1 *Data Packaging*

The transport layer entity will package data coming from the user application (file transfer for instance) into a suitable format called a Transport Protocol Data Unit (**TPDU**). This package will include the Destination Address and additional control information.

5.1.2 *Connection Management*

Again it is considered that there is a **"virtual link"** between layer 4 entities. Layer 4 has the responsibility for setting up, maintaining, and terminating the layer virtual link. For this purpose it will use the same kind of tools as layer 3 (commands and responses).

5.1.3 *Status Reporting*

Transport layer will report any irrecoverable error to the next upper layer.

5.2 TRANSPORT ADDRESSES

The Transport layer defines a set of transport addresses which generally consists of a Network number, a Host number and a Port number attached to the Host.

CCITT for example specifies a 14 digit numbering scheme (Recommendation X.121) in which 4 digits identify the Network and 10 digits identify the Host and its port.

A typical use of this numbering system would be:

country	net	host	port

Note that these transport numbers have no relation to the network (or logical) number. For example, the transport connection can use many virtual circuits for the purpose of transferring a single message **(downward multiplexing)** or the opposite, multiplex many transport data units on a single layer 3 virtual circuit **(upward circuit multiplexing)**.

Flow Control implementation at the Transport level is far more complex than at lower layers for two reasons:

First because the data-link between Transport entities is only virtual (it is a long way for an acknowledgement to get through).

Secondly, because of the long and often variable delays involved on a round trip.

As the Transport layer software implementation usually uses a limited amount of storage it consequently must implement a "no more, I have enough" mechanism to keep up with the incoming data flow.

5.2.1 *Credit Allocation*

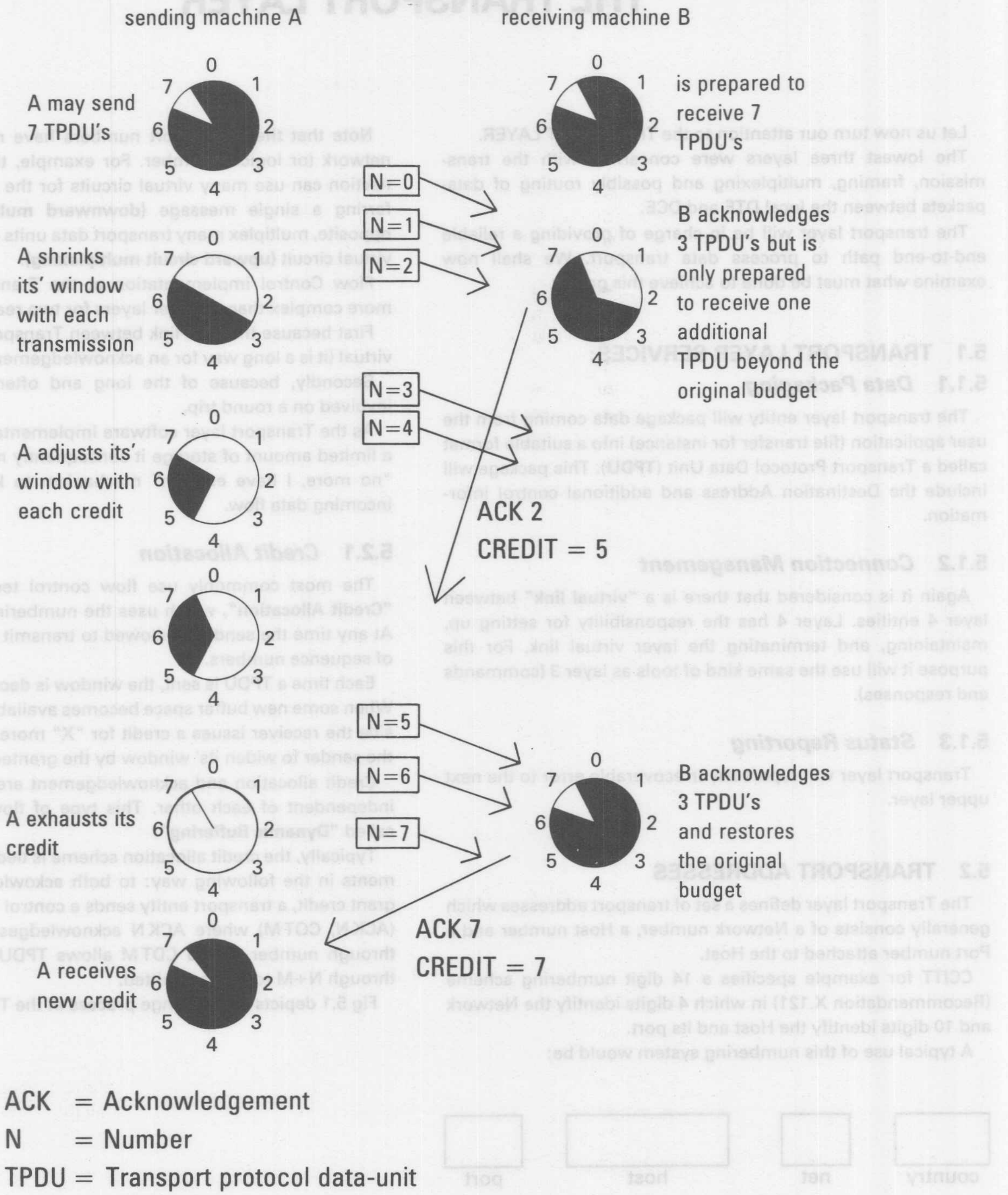
The most commonly use flow control technique is called **"Credit Allocation"**, which uses the numbering of the TPDU's. At any time the sender is allowed to transmit only a "window" of sequence numbers.

Each time a TPDU is sent, the window is decremented by one. When some new buffer space becomes available on the receiver side the receiver issues a credit for "X" more TPDU's allowing the sender to widen its' window by the granted amount.

Credit allocation and acknowledgement are thus completely independent of each other. This type of flow control is also called **"Dynamic Buffering"**.

Typically, the credit allocation scheme is tied to acknowledgements in the following way: to both ackowledge TPDU's and grant credit, a transport entity sends a control TPDU of the form (ACK N, COT M) where ACK N acknowledges all data TPDU's through number N and CDT M allows TPDU's numbers N+1 through N+M to be transmitted.

Fig 5.1 depicts an exchange process at the Transport level.



In this example of a credit allocation protocol, data flows in one direction. Sending Machine A is granted a credit allocation of 7.

CHAPTER 6

THE SESSION, PRESENTATION, and APPLICATION LAYERS

6.1 INTRODUCTION

Layer 1 to 4 provided the user with an end-to-end delivery facility. Chunks of bits sent from Host A have reached their destination, Host B, in an orderly fashion. The problem now is to make these data usable by the destination processes.

Again to take an imaged example, the telephone operating company can provide you with a good quality telephone connection to a Chinese correspondent, but this is useless if you cannot speak Chinese or a common language.

The purpose of layers 5 to 7 will be to process the data received in order to render them useful and also to provide some specific synchronisation and bookeeping facilities.

Let us now review each of these layers in detail.

6.2 THE SESSION LAYER

The Session Layer will provide the following services to the Transport Layer:

★ Reliable and Transparent Data Transfer

★ Organised Data Transfer

★ Synchronised Data Transfer

Reliable and transparent data transfer are derived directly from the Transport layer, the other two characteristics are added values provided by the session layer.

The main purpose of the Session layer is generally to make Transport connection failures transparent to the higher layers.

Synchronisation service is used for Data-base updates. The updating information is usually saved in a buffer until the update is complete. It is only then that the actual update is done in order to prevent accidental network failures causing permanent damage to the Data-base.

6.3 THE PRESENTATION LAYER

The main goal of this layer is to format the data in order to make them understandable to their destination. For example, ASCII data may need to be converted to EBCDIC.

It is also the task of layer 6 to handle problems such as Terminal interfacing. Today there are hundreds of different brands of terminals on the market. When you are doing some remote log-in procedure, you would not like the remote host telling you that it cannot work with your terminal.

Special programs, generally called virtual terminal protocols, have been designed to cope with these problems. In the OSI model, they are classified in the services offered by the presentation layer.

Another service that could be part of the Presentation layer is a data Encryption/Decryption facility.

Data Encryption/Decryption can also be done by special Harware at the link level. In this case, it is called Data link encryption.

The two methods are different. With Data link encryption, the choice of the coding method is fixed on the installation, whereas in a end-to-end encryption done at the Presentation layer, the encoding method can be agreed upon before the communication takes place, and can be changed for the next session.

A special class of program called **"Virtual file protocol"** is also found in the Presentation layer. The situation here is comparable to the "Virtual Terminal" problem. If we want to transfer files between different vendor's equipment we have to make sure that the receiving computer understands the file structure it is receiving, (records organisation, length, etc...) in order to store it correctly in its' own mass storage according to its' operating system requirements.

6.4 THE APPLICATION LAYER

This is the highest layer of the OSI model. In this way it is different from the others because it does not provide any "service" to an upper layer.

Generally speaking the Application layer will include network oriented utilities that are not intrinsic to the local host's operating system.

A typical large class of programs belonging to the Application layer is constituted by the distributed data base software (e.g. used by airline offices for worldwide ticket reservation).

The Application layer also includes what is called now "Network operating systems" with all the **"Servers"** that allow a particular user to have access to remote facilities (remote printers, hard disks, etc...).

GENERAL BIBLIOGRAPHY

CCITT

★ ISDN
STUDY GROUP XI REPORT P 58
DIGITAL ACCESS SIGNALLING SYSTEM Q.920–931
STUDY GROUP XVIII REPORT R32

★ **SERIES OF RECOMMENDATIONS**
RECOMMENDATION X.25
STUDY GROUP VII REPORT R 31
RECOMMENDATIONS X.26–X.57
STUDY GROUP VII REPORT R 32
RECOMMENDATIONS X.60–X.75

ISO

★ ISO 4335 DATA COMMUNICATION HIGH LEVEL DATA LINK CONTROL PROCEDURES, ELEMENT OF PROCEDURE AND APPENDICE 1, 2.

★ ISO 3309 HIGH LEVEL DATA LINK CONTROL PROCEDURES, FRAME STRUCTURES.

★ ISO TC/97 SCG 2372–929 NETWORK SERVICE DEFINITION. 861 BASIC TRANSFER PROTOCOL SPECIFICATION.

★ ISO 7498 MODELE DE REFERENCE DE BASE.

★ ISO 7776 X.25 LAPB COMPATIBLE DTE LINK PROCEDURES.

ECMA STANDARD ECMA

40 HDLC FRAME STRUCTURE	JAN 80
49 HDLC ELEMENTS OF PROCEDURE	APR 79
60 HDLC UNBALANCED CLASS OF PROCEDURE	APR 79
61 HDLC BALANCED CLASS OF PROCEDURE	APR 79
71 HDLC SELECT PROCEDURES	JAN 81
72 TRANSPORT PROTOCOL	SEP 82
75 SESSION PROTOCOL	JAN 82
TR 16 INTERFACE CHARACTERISTICS FOR A DTE TO OPERATE WITH EUROPEAN REC. X.25 NETWORKS	SEP 83
TR 15 ANALYSIS OF EUROPEAN X.25 NETWORKS	SEP 85

Published Articles:

DATA COMMUNICATIONS

★ TESTING OSI PROTOCOLS: NBS ADVANCES THE STATE OF THE ART
MAR 84 277–285

★ OF LOCAL NETWORKS, PROTOCOLS AND OSI REFERENCE MODEL
NOV 84 129–150

★ A PRIMER: UNDERSTANDING TRANSPORT PROTOCOL
NOV 84 205–215

★ A GUIDE TO DATA COMMUNICATION SOFTWARE OPERATIONAL CHARACTERISTICS: BCS VERSUS SDLC
NOV 83 121–158

★ PACKET SWITCHING AND X.25, WHERE TO FROM HERE. HOW MUCH SUPPORT FOR COMMUNICATIONS DO MICROCOMPUTER VENDORS OFFER
OCT 83 121–138

★ MODEMS: SIMPLE TECHNIQUES FOR ISOLATING COMMON FAULTS
SEP 83 107–110

★ KEEPING TRANSACTIONAL NETWORKS CLEAN: THE KEY CRITERIA
SEP 83 161–167

★ HDLC, ADCCP AND SDLC: WHAT IS THE DIFFERENCE
AUG 83 115–136

★ THE STATE OF AFFAIRS IN WORLDWIDE PACKET NETWORKS
FEB 83 97–100

★ HIGH LEVEL PROTOCOLS, STANDARDS AND THE OSI MODEL
JUL 82 71–101

★ REALITY AND THE PROPOSED OSI STANDARD
JUN 81 77–80

Motorola reserves the right to make changes without further notice to any products herein to improve reliability, function or design. Motorola does not assume any liability arising out of the application or use of any product or circuit described herein; neither does it convey any license under its patent rights nor the rights of others. Motorola and Ⓜ are registered trademarks of Motorola, Inc.